Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

TC 11 briefing papers

A novel cryptosystem based on DNA cryptography and randomly generated mealy machine

Pramod Pavithran[a], *Sheena Mathew*[a], *Suyel Namasudra*[b,*], *Pascal Lorenz*[c]

[a] *Division of Computer Engineering, School of Engineering, Cochin University of Science and Technology, Kochi, India*
[b] *Department of Computer Science and Engineering, National Institute of Technology Patna, Bihar, India*
[c] *University of Haute-Alsace, Colmar, France*

ARTICLE INFO

Article history:
Received 17 September 2020
Revised 20 December 2020
Accepted 21 December 2020
Available online 25 December 2020

Keywords:
DNA cryptography
DNA sequence
Encryption
Decryption
Mealy machine

ABSTRACT

Nowadays, the amount of data produced and stored in computing devices is increasing at an alarming rate. Tremendous amounts of critical and sensitive data are transmitted between all these devices. Thus, it is very imperative to guarantee the security of all these indispensable data. Cryptography is a commonly used technique to ensure data security. The fundamental objective of cryptography is to transmit data from the sender to the receiver in the most secure way, so that an attacker is unable to extract the original data content. This paper proposes a novel cryptosystem based on Deoxyribonucleic Acid (DNA) cryptography and finite automata theory. The system is made of three entities, namely a key pair generator, a sender and a receiver. The sender generates a 256-bit DNA based secret key based on the attributes of the receiver, and this key is used for data encryption. Then, a randomly generated Mealy machine is used for coding the DNA sequence, which makes the ciphertext more secure. The proposed scheme can protect the system against numerous security attacks, such as brute force attack, known plaintext attack, differential cryptanalysis attack, cipher text only attack, man-in-the-middle attack and phishing attack. The results and discussions show that the proposed scheme is efficient and secure than the existing schemes.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Cryptography is considered as the most prominent solution for ensuring data security. The main aim of cryptography is to transmit data from the sender to the receiver over an insecure communication medium in a disguised form such that an attacker is unable to read the original data. Cryptography is broadly classified into symmetric key encryption and asymmetric key encryption. In symmetric key encryption, the same key is used for the encryption and decryption process (Kumar and Wollinger, 2006; Schneier, 1996), whereas in asymmetric key encryption, two different keys are used for encryption and decryption processes. These keys are known as the private key and public key. The private key is kept as secret, whereas the public key is made available to all.

* Corresponding author.
E-mail address: suyelnamasudra@gmail.com (S. Namasudra).
https://doi.org/10.1016/j.cose.2020.102160
0167-4048/© 2020 Elsevier Ltd. All rights reserved.

DNA cryptography is an emerging field in cryptography, which has emerged from the progress of DNA computing (Sundaram et al., 2015). Unlike traditional cryptography, DNA cryptography utilizes the characteristics of DNA along with cryptographic techniques to ensure the security of data. Rapid changes in the field of information technology and vast volumes of data generated every day, makes the security of data a major concern. Nowadays, DNA cryptography is widely used in the field of cryptography. Many researchers have proposed several schemes based on DNA cryptography to improve data or information security. All these schemes are found to exhibit security issues. Moreover, some schemes take excessive time for secret key generation and key retrieval. The time taken for encryption and decryption are also not satisfactory as these are high.

In the last few years, many researchers have implemented various cryptographic algorithms for efficient and secure data transmission (Namasudra et al., 2020a; Li et al., 2019; Namasudra et al., 2020b; Alguliyev et al., 2020; Namasudra et al., 2017; Zhao et al., 2019; Namasudra and Roy, 2017; Tornea and Borda, 2013; Namasudra et al., 2020c; Hsu and Lee, 2006; Namasudra, 2019; Gehani et al., 2000; Wang and Zhang, 2009; Namasudra et al., 2020d; Sarkar et al., 2015; Clelland et al., 1999; Namasudra, 2018a; Zhang and Gao, 2015; Gupta and Singh, 2015; Devi et al., 2020; Leier et al., 2000; Namasudra, 2018b; Wang and Yu, 2011; Namasudra et al., 2018; Zhang et al., 2012; Namasudra et al., 2020e; Namasudra, 2020a; Namasudra and Deka, 2021). Some of the implementations are reviewed in this section. Rama et al. (2012) have made a study of the Data Encryption Standard (DES) algorithm using Cellular Automata (CA). In this scheme, there must have the ability to produce random keys during data accessing, and the use of cellular automata rule is easy to apply for the operation of the DES algorithm. The CA rule number 30 is used for key generation. This rule is shown for processing randomness, but the key size used is only 48 bits. Sharma et al. (2013) have proposed a technique for secure transmission of the text. This technique aims to provide two levels of security. In the first level, an image is used for hiding the text using a password, and in the second level, two-dimensional cellular rules are used for further encryption. In case there is a breach in the first level of security, then the second level of protection ensures more safety to the transmitted message. Two-dimensional rules of cellular automata are used to perform encryption. The limitation of this technique is the high key generation time. Sony et al. (2013) have proposed a novel algorithm for DNA cryptography based on the concept of the Moore machine in the automata theory. The security of this technique relies on three encryption stages, namely the use of a secret key, auto-generated Moore machine and a password. In the algorithm of Sony et al., the message is first encrypted by using a secret key that is dynamically generated. The encrypted data is converted into a binary sequence, which is partitioned into 'n' blocks of 256-bit each. The Exclusive OR (EXOR) operation is performed on the blocks, and the results of EXOR is converted into a DNA sequence using a codebook (lookup table). The DNA sequence is given to a Moore machine that produces a user's password as output. The newly created password is again converted to a DNA sequence using a codebook. This DNA sequence is the final encrypted data that is sent to the receiver. However, this algorithm is very slow.

Hossain et al. (2016) have proposed a new technique for DNA cryptography that uses a DNA sequence table to enhance the level of security. Initially, the random American Standard Code for Information Interchange (ASCII) characters are assigned to a DNA sequence table. During each data transmission process, a finite number of iterations are applied on a mathematical series, where the position of the ASCII characters are changed dynamically in the sequence table. Then, a One-Time-Pad (OTP) is applied to the modified binary value. Again, the OTP ciphertext is processed through genomic conversion. Finally, it is converted into a compressed ciphertext using an amino acid table consisting protein sequence that increases the confusion in the ciphertext. Kalyani and Gulati (2016) have proposed a new method for secure data transmission using a three-level security, namely biological process (DNA complementary rule), arithmetic operation (EXOR operation) and OTP scheme. The data is first converted into a binary sequence. The binary sequence is then EXORed with an OTP. The resultant binary sequence is converted into a DNA sequence as per the rules: 00=*A*, 01=*T*, 10=*C* and 11=*G*. Then, the elementary rule is applied to get a new DNA sequence. The sequence is reversed to get the final encrypted data, which is sent to the receiver.

To solve some of the issues of the above-mentioned schemes, a novel approach for data encryption is proposed in this paper. The proposed scheme makes use of the DNA cryptography and automata theory. Here, both the sender and receiver receive a key pair (public key and private key) from a key pair generator. These keys are used for the secure transmission of the ciphertext and other parameters between the sender and the receiver. The paper can be summarized by the following main contributions:

1) A novel encryption technique has been proposed in this paper by using a 256-bit secret key, DNA coding and Mealy machine.
2) To improve data security, a key-pair generator is used to generate public and private keys for the sender and receiver. This is to ensure secure exchanging of parameters between the sender and the receiver.
3) Experimental results are presented in this paper to prove that the proposed scheme is more efficient than the existing scheme.

The sections to follow in the paper are the background of the proposed scheme in Section 2 followed by the proposed scheme in Section 3. A case study of the proposed scheme is given in Section 4 followed by the security analysis in Section 5. The complexity analysis with compare other existing schemes has been presented in Section 6. The performance analysis of the proposed scheme has been discussed in Section 7, and finally, Section 8 represents conclusions and future works.

2. Background of the proposed scheme

This section deals with two main aspects of the proposed scheme.

2.1. DNA cryptography

DNA is made of molecules that are known as nucleotides. Each nucleotide consists of a single base, and there are four kinds of bases, which are named as Adenine (A), Thymine (T), Cytosine (C) and Guanine (G). In DNA based encryption, data are encrypted and stored by using the DNA bases, namely A, T, C and G, unlike the conventional approaches, where binary bits i.e. 0 and 1 are used for data encryption. So, the sender selects one of the combinations of DNA bases for encrypting the data, which helps to improve data security. A combination of DNA bases is known as the DNA sequence. Hence, the data is encrypted and converted into a DNA sequence. For example, let the encoding formula for the DNA bases is 00 → A, 01→ T, 10 → C and 11 → G and consider the encryption of character 'H'. If the binary value of 'H' is equal to 01001000, by using the DNA encoding, the character is represented as 'TACA'.

2.2. Mealy machine

The Mealy machine is a finite state machine (Hopcroft et al., 2008), whose output depends on the current state of the machine and the current input. It is defined by six tuples i.e., $M = (Q, \Sigma, O, \delta, f, q_0)$. where,

Q	-	A non-empty finite set of states
Σ	-	A non-empty finite set of input symbols
O	-	A finite set of output symbols
δ	-	Input transition function, where $\delta : Q x \sum \rightarrow Q$
f	-	Output transition function, where $f : Q x \sum \rightarrow O$
q_0	-	Initial state

The Mealy machine is randomly and dynamically generated for each data transmission. In the proposed scheme, the role of the Mealy machine is to read each DNA base presented in a DNA sequence and to make a substitution with another DNA base.

2.3. Design of dynamic random mealy machine (DRMM)

The DRMM is designed with four states, where '0′ is the initial state of the machine. The DRMM is defined as $M = (Q, \Sigma, O, \delta, f, q_0)$. where,

Q	-	{0, 1, 2, 3}
Σ	-	{A, T, C, G}
O	-	{A, T, C, G}
δ	-	Randomly generated
f	-	Randomly generated
q_0	-	0

2.3.1. Transition tables

The design of the DRMM is mainly represented using two transition tables, namely the state table and output table. The state table stores the mapping function $\delta : Q x \sum \rightarrow Q$ as shown in Table 1 and the output table stores the mapping function $f : Q x \sum \rightarrow O$ as shown in Table 2. The values of 'next states' in Table 1 are assigned randomly from the set {0, 1, 2, 3}. The 'output' values of Table 2 are assigned randomly from the set {A, T, C, G}.

Table 1 – State table.

State	Input 'A' Next state	Input 'T' Next state	Input 'C' Next state	Input 'G' Next state
0	1	2	0	3
1	3	1	2	0
2	0	3	1	2
3	2	0	3	1

Table 2 – Output table.

State	Input 'A' Output	Input 'T' Output	Input 'C' Output	Input 'G' Output
0	T	A	G	C
1	C	T	A	G
2	G	C	T	A
3	A	G	C	T

2.3.2. Transition diagram

The state table and output table generated in Section 2.3.1 are used to create the Mealy machine, which is represented as a transition diagram shown in Fig. 1. Circles represent the states of the Mealy machine. Here, '0′ is the initial state of the Mealy machine. Processing starts from the initial state of the Mealy machine. A labelled directed arrow represents the transition of the Mealy machine. Consider the first DNA base in the DNA sequence is 'A'. The DNA base 'A' is converted to the DNA base 'T' by the Mealy machine and the machine makes a transition from state 0 to state 1. The notation, 'A/T' (input/output) labelled on the arrow from the state '0′ to the state '1′ represents the input-output translation. Another transition occurs from state '1′, thereby giving another DNA base as the output. The process continues until all the DNA bases in the DNA sequence are completed. In this way, any DNA sequence given as input to the Mealy machine is converted into another new DNA sequence.

3. Proposed scheme

The objective of the proposed scheme is to develop a highly secured system for data transmission from a sender to a receiver. There are three entities in the proposed system:

1) *Key-Pair Generator (KPG)*: This entity provides the public and private keys to the sender and receiver during the registration process.
2) *Sender*: This entity stores confidential or normal data.
3) *Receiver*: This entity sends requests to the sender for accessing any required data

At first, the KPG provides the key-pairs to the sender and receiver on request. The keys are used for exchanging confidential parameters needed for encrypting and decrypting

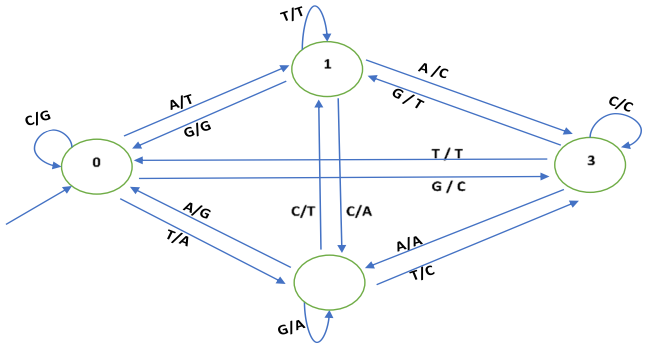

Fig. 1 – Mealy machine design.

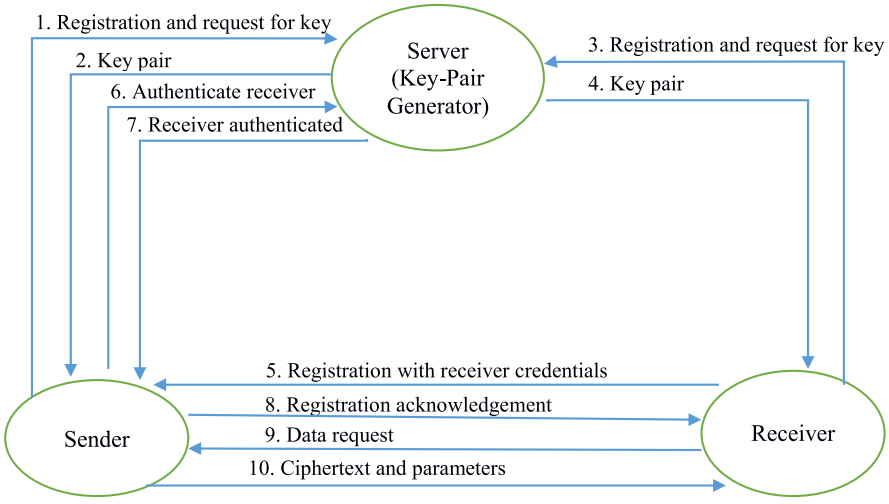

Fig. 2 – Workflow of the proposed scheme.

the data. The receiver first sends a registration request to the sender along with a few attributes of the receiver. The sender checks the authenticity of the receiver from the KPG. If the receiver is authenticated, the sender proceeds with the generation of a 256-bit Receiver Attribute Secret Key (RASK). After encrypting the data with RASK, the encrypted data is converted into a DNA sequence. The DNA sequence is applied to a randomly generated Mealy machine, which transforms the input to a new DNA sequence, and finally, generates the ciphertext. The proposed system is composed of few phases: (i) system setup (ii) receiver registration (iii) key generation (iv) generation of DNA based Mealy machine (v) data encryption, and (vi) data decryption. Fig. 2 shows the workflow of the entire system.

3.1. *System setup phase*

The sender and receiver send a registration request to the KPG. The KPG generates public and private key-pairs for the sender and receiver. The key-pairs are given to the entities, where the private key is kept secret from the other entity.

3.2. *Receiver registration*

The receiver sends a registration request to the sender along with some personal attributes of the receiver, such as Personal Account Number (PAN), email Identity (ID) and the Media Access Control (MAC) address. The attributes are sent in an encrypted form using the receiver's private key and sender's public key. The sender confirms the authenticity of the receiver from the KPG. If authenticated, an acknowledgment is sent to the receiver.

3.3. *Key generation*

The sender performs the key generation process based on the attributes collected from the receiver. The attribute values received in the encrypted form are decrypted using the sender's private key and receiver's public key. The values are converted into the corresponding ASCII values by using Table 3. Subsequently, the resultant ASCII values are replaced by their equivalent binary values. If the resultant binary number is less than 256-bit, extra zeroes are padded at the end of the binary number. Otherwise, extra bits are truncated from the end of the binary number. Thus, a 256-bit secret key (RASK) is generated. The steps for generating the RASK are as follows:

Table 3 – Encoding of ASCII values as 8-bit combinations.

Character	ASCII value	Binary code
0	48	10011100
1	49	00111001
2	50	11000110
3	51	01100011

Step 1: In the first step, the sender accepts the receiver attributes i.e. email ID, MAC address and PAN.

Step 2: Concatenate email ID and MAC address to form a string.

Step 3: Each character of the string obtained in step 2 is converted to its equivalent ASCII value.

Step 4: In the fourth step, each character of PAN is converted into its ASCII value.

Step 5: Merge the ASCII values obtained in step 3 and step 4 as given below.

⟨*e*|*e*|*e*|*e*|*e*|*e*|*e*|*e*|*e*|*e*|*e*|*e*⟩ and ⟨***p***|***p***|***p***|***p***⟩ are merged as below: ⟨***p***|*e*|***p***|*e*|***p***|*e*|***p***|*e*|***p***|*e*|*e*|*e*|*e*|*e*|*e*|*e*⟩

Step 6: Convert the merged ASCII values into the binary string.

Step 7: Finally, the string of step 6 is converted into a 256-bit binary string (RASK) by proper padding with '0's or removing extra bits from the right end of the string.

3.4. *Generation of DNA based Mealy machine (DNAMM)*

The sender generates the state table and output table as mentioned in Section 2.3.1. The steps for generating the DNAMM are as follows:

Step 1: Convert the 16 DNA bases of the output table into characters 0, 1, 2, and 3. i.e. A, T, C, and G are replaced by characters 0, 1, 2 and 3, respectively.

Step 2: Copy the 16 characters from both the state table and output table column-wise to form a string (32 characters).

Step 3: In the third step, each character of the string is converted into its equivalent ASCII value by using Table 3.

Step 4: Convert the ASCII values to 8-bit binary values, using Table 3 (randomly generated with equal number of 0's and 1's), to form a 256-bit binary number.

Step 5: In the fifth step, each two bits of the binary number obtained in step 4 are converted into the corresponding DNA base as per the Row Number (R1) randomly selected from Table 4. Finally, the state table and output table are coded as a 128 base DNA sequence (DNAMM).

The DNAMM, Table 3 and R1 are used by the receiver to retrieve the state table and output table.

3.5. *Data encryption*

The sender selects the plaintext and converts each character of the plaintext into its ASCII value. Then, each ASCII value is converted to its equivalent binary value to form a binary string. If the Length (len) of the binary string is greater than 256-bit, the binary string is divided into 256-bit blocks. EXOR operation is then performed between each 256-bit block and the 256-bit RASK. If the last block is less than 256-bit, truncates the RASK and performs EXOR operation with the last block. Then, each two bit of the binary string is converted to its corresponding DNA base as per the Row (R2) selected from Table 4. Let this DNA sequence is denoted as dnaD1. Each DNA base of dnaD1 is given to the Mealy machine as input. The Mealy machine converts dnaD1 to a new DNA sequence i.e. dnaD2 by using transition. Then, dnaD2 is reversed to generate another DNA sequence, which is the final ciphertext.

After encryption, the ciphertext, DNAMM, Table 3, parameters R1 and R2 are encrypted using the sender's private key and receiver's public key, and then, it is sent to the receiver. The algorithm for encrypting the plaintext using RASK is given as Algorithm 1.

Table 4 – DNA bases for 2-bit binary.

Row	00	01	10	11
0	A	T	C	G
1	A	C	T	G
2	C	T	A	G
3	C	T	G	A
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
22	G	T	A	C
23	G	T	C	A

Algorithm 1 – Encryption Algorithm.

Input: Plaintext
Output: Ciphertext

1. START
2. CONVERT each character of plaintext to ASCII values
3. CONVERT ASCII values to their equivalent binary string
4. IF len $\geq$ 256 bits
5. DIVIDE binary string into 256-bit blocks
6. EXECUTE EXOR between each 256-bit block and 256-bit RASK
7. IF len $<$ 256 bits
8. EXECUTE EXOR between truncated RASK and binary string
9. SELECT R2 from Table 4
10. PERFORM DNA coding on the binary string to generate dnaD1
11. FOR each DNA base in dnaD1
12. APPLY transition rule to find the next state and the new DNA base
13. APPEND new DNA base to dnaD2
14. end FOR
15. REVERSE dnaD2
16. END

3.6. *Data decryption*

The receiver decrypts the message using the receiver's private key and sender's public key to recover the ciphertext, DNAMM, Table 3, parameters R1 and R2. At first, the receiver reverses the ciphertext to form the DNA sequence i.e., dnaD2. Then, DNAMM is decoded by performing the steps of Section 3.4 in the reverse order. Thus, the state table and output table of the Mealy machine are regenerated. The receiver has also retrieved the 256-bit secret key (RASK) using his/her attributes.

Algorithm 2 – Decryption Algorithm.

Input: Ciphertext
Output: Plaintext

1. START
2. REVERSE the ciphertext
3. DECODE DNAMM
4. FOR each DNA base in dnaD2
5. APPLY Tables 1 and 2 to find the next state and the new DNA base
6. APPEND new DNA base to dnaD1
7. end FOR
8. CONVERT dnaD1 to binary string
9. IF len $\geq$ 256 bits
10. DIVIDE binary string into 256-bit blocks
11. EXECUTE EXOR between each 256-bit block and 256-bit RASK
12. IF len $<$ 256 bits
13. EXECUTE EXOR between truncated RASK and binary block
14. CONVERT binary string to equivalent ASCII values
15. CONVERT ASCII values to plaintext
16. END

Each DNA base from dnaD2 is given to the Mealy machine as input. When all the DNA bases are given to the Mealy machine, a new DNA sequence i.e., dnaD1 is generated by applying the transition rules of the Mealy machine in the reverse order. Then, each DNA base of dnaD1 is replaced by its equivalent 2-bit binary value obtained from row (R1) of Table 4. If len is greater than 256-bit, the binary string is divided into 256-bit blocks. EXOR operation is performed between each 256-bit block and 256-bit RASK. If the last block is less than 256-bit, truncates the RASK and performs the EXOR operation with the last block. At last, the binary string is replaced by its equivalent ASCII values, which is further converted to the original plaintext. The algorithm for decrypting the ciphertext using RASK is presented as Algorithm 2.

4. Case study of the proposed scheme

This section illustrates a sample execution of the proposed scheme.

4.1. *Key generation*

The sender executes the following steps to generate the key for encrypting the plaintext:

a) The sender gets the receiver's attributes i.e., email ID, MAC address and PAN as: abcdefgh@gmail.com, 0a-12-22-b4-99-66, aeypp8436r
b) Concatenate the strings represent the email ID and MAC address.

abedefgh@gmail.com0a-12-22-b4-99-66

c) Convert each character of the string to its equivalent ASCII value.

97 98 99 100 101 102 103 104 64 103 109 97 105 108 46 99 111 109 48 97 49 50 50 98 52 51 57 57 54 54

d) Convert each character of PAN to its ASCII value.

97 101 121 112 112 56 52 51 54 114

e) Merge the ASCII values obtained in (c) and (d) as given below:

97 97 **101** 98 **121** 99 **112** 100 **112** 101 **56** 102 **52** 103 **51** 104 **54** 64 **114** 103 109 97 105 108 46 99 111 109 48 97 49 50 50 98 52 51 57 57 54 54

f) Convert the value of (e) to the binary string.

011000010110000101100101011000100111100101100011011100000110001000111000001100101001110000110011000100001110000011001010011100001100110001101000100000000110011011001110011011001101110101110010011000010110100101101100001011100110001101101111011011010011000100100110000000110011001100110011010100111000001101010011001110011000000011000 10011010 00011010 (304 bits)

g) Convert the binary string of (f) to a 256-bit binary string (RASK) by removing extra bits from the right end of the string:

011000010110000101100101011000100111100101100011011100000110001000111000001100101001110000110011000100001110000011001010011100001100110001101000100000000110011011001110011011001101110101110010011000010110100101101100001011100110001101101111011011010011000100100110000000110011001100110011010100111000 (256 bits)

4.2. *Coding Mealy machine as DNA sequence*

The sender generates the DNA sequence based on Mealy machine by executing the following steps:

a) Copy the assigned 16 characters from both the state table and output table to form a 32-character string as given below:

1 3 0 2 2 1 3 0 0 2 1 3 3 0 2 1 3 0 2 1 1 3 0 2 2 1 3 0 0 2 1 3

b) Convert each character of the string to its equivalent ASCII value by using Table 3.

49 51 48 50 50 48 51 48 48 50 49 51 51 48 50 49 51 48 50 49 49 51 48 50 50 49 51 48 48 50 49 51

c) Convert each ASCII value to its equivalent 8-bit binary value.

00110001100011100110011000110110001101001110001100011100110011000110001110010011100011000110001110010110001110011100110001101100011000111000110001100011100101100011100110011000110001110010011100011000110001110011100110001101100011000111001100011100101100011100110011000110001110010011100011000110001110011100110001101100011000111001011000111001100110001100011100100111000110

d) Replace each two bits of the binary number obtained in (c) by the corresponding DNA base as per the row number of Table 4.

AGCTTCAGCTGAGATCGATCCTGATCAGCTGACTGAGATCA
GCTTCAGTCAGCTGAGATCAGCTTCAGCTGAGATCAGCTAGCT
TCAGCTGAGATCGATCAGCTTCAGCTGACTGAGATCAGCTTCA
G (128 bases)

4.3. *Data encryption*

The data encryption process is executed by the sender by performing the following steps:

a) The sender selects the plaintext 'hello'.
b) Convert each character of plaintext to its ASCII value.

104 101 108 108 111

c) Convert ASCII values to its equivalent binary value to form a binary string.

0110100001100001011011000110110001101111 (40 bits)

d) Since the length is less than 256, RASK is truncated by taking the last 40 bits as given below:

0011000000110011001100110011010100111000 (40 bits)

e) EXOR operation is performed between the binary string and the truncated RASK.

0101100001010010010111110101100101010111 (40 bits)

f) Convert each two bits of the binary number obtained from (e) to the corresponding DNA base as per row number 0 of Table 4.

dnaD1 = TTCATTACTTGGTTCTTTTG

g) dnaD1 is given as input to the Mealy machine to generate a new DNA sequence i.e., dnaD2.

dnaD2 = ACCACGTACGCTTTACGACT

h) Reverse dnaD2 to generate another DNA sequence, which is the final ciphertext.

TCAGCATTTCGCATGCACCA

4.4. *Data decryption*

The receiver executes the following steps for data decryption:

a) The receiver reverses the ciphertext TCAGCATTTCGCATGCACCA to get a DNA sequence i.e. dnaD2.

dnaD2 = ACCACGTACGCTTTACGACT

b) Then, dnaD2 is given as input to the Mealy machine to generate another DNA sequence i.e. dnaD1. This is the reverse process of 'step g' of subSection 4.3. Here, the transition rules of the Mealy machine are executed in the reverse order. The entire process is described in detail as follows:
   1) At first, the initial state is 0 and the first DNA base of dnaD2 i.e. 'A' is considered.
   2) In the output table (Table 2), 'A' is presented in column input as 'T' for state 0. So, 'A' is replaced by 'T' and 'T' is appended to dnaD1. Now, dnaD1 is 'T'.
   3) In the state table (Table 1), for state 0 and for column input 'T', the state changes from 0 to 2 i.e. state = 2.
   4) Now, consider the second DNA base of dnaD2 i.e. 'C' and the state is 2.
   5) In the output table (Table 2), 'C' is presented in column input as 'T' for state 2. Therefore, 'C' is replaced by 'T' and 'T' is appended to dnaD1. Thus, the updated dnaD1 is 'TT'.
   6) In the state table (Table 1), for state 2 and for column input 'T', the state changes from 2 to 3 i.e. state = 3.
   7) The above mentioned processes are continued for the remaining DNA bases of dnaD2, and finally, the original DNA sequence i.e. dnaD1 = TTCATTACTTGGTTCTTTTG is generated by the Moore machine.
c) dnaD1 is converted into its equivalent binary number as per the row number 0 of Table 4.

0101100001010010010111110101100101010111 (40-bit)

d) Since the length is less than 256-bit, RASK is truncated by taking the last 40 bits as given below:

0011000000110011001100110011010100111000 (40-bit)

e) EXOR operation is performed between the resultants of the above two steps.

0110100001100001011011000110110001101111 (40-bit)

f) The binary number is converted into its equivalent ASCII values.

104 101 108 108 111

g) At last, the ASCII values are converted to their equivalent characters to retrieve the plaintext 'hello'.

5. Security analysis

The security analysis of the proposed scheme is presented in this section.

5.1. *Brute-force attack*

In the brute-force attack (Salamatian et al., 2019), the intruder tries to guess the secret key used for encryption. In the proposed scheme, the encryption process takes place using a 256-bit secret key. An attacker has to make 2^{256} guesses to hack the key, which is computationally impossible. Hence, the proposed system is found resilient to brute-force attacks.

5.2. *Known plaintext attack*

During known plaintext attack (Yuan and Mo, 2020), the attacker tries to access the ciphertext and its corresponding plaintext. The goal is to guess the secret keys and to develop a technique that can be used for decrypting any further messages. Here, in the proposed scheme, EXOR operation is performed between each two bits of the plaintext and the corresponding bits of the secret key to generate a new binary string. Further, the binary string is replaced by the DNA bases as per the random DNA coding rule of Table 4. Hence, any two similar plaintexts are encrypted to make two ciphertexts, which are entirely different. Thus, the system is able to withstand this type of attack.

5.3. *Ciphertext only attack*

In the ciphertext only attack (Chang et al., 2020), the attacker tries to access only several encrypted messages. The main aim of this attack is to recover as many plaintext messages as possible or to guess the secret keys. Once the encryption key is hacked, decrypting all the messages encrypted using this key is possible. In the proposed scheme, the transition tables (Tables 1 and 2) used in the Mealy machine are randomly generated for every data transmission. So, the DNA sequences generated for two similar ciphertexts during decryption are completely different. Therefore, the plaintexts are also different, and thus, the proposed scheme can resist this attack.

5.4. *Differential cryptanalysis attack*

A differential cryptanalysis attack (Zhao et al., 2020) is considered as a method of analysing pairs of plaintext and corresponding ciphertext to hack the key or more precisely to minimize the time needed to get the key. The proposed scheme is secured against this attack as a new secret key is generated by using different attributes of the user and a randomly generated Mealy machine. The same plaintext gets converted into varying ciphertexts during each transmission, and thus, the proposed scheme improves security.

5.5. *Man-in-the-middle attack*

In the man-in-the-middle attack (Nath Nayak and Samaddar, 2010), the attacker intercepts communications between the sender and receiver. This is either eavesdrop or modify the traffic between two entities. Attackers use this attack to get personal information, spy on people, disrupt communications or corrupt data. In the proposed scheme, the personal attributes of the receiver are encrypted using the receiver's private key and the sender's public key before it is transmitted to the sender. Similarly, the ciphertext, DNAMM, Table 3, parameters R1 and R2 are encrypted using the sender's private key and the receiver's public key before they are sent to the receiver. Therefore, the intended receiver only can retrieve the details by using own private key and the sender's public key. In the proposed scheme, both the keys are secret and are not shared with anyone except the intended one. Thus, the proposed scheme resists this attack.

5.6. *Phishing attack*

In the phishing attack (Baykara and Gürel, 2018; Namasudra, 2020b), a malicious user or unauthorized user tries to retrieve the personal information of the receiver. The receiver's sensitive information can be used by the unauthorized user to access the sender's data. In the proposed scheme, the sender and the receiver have to register with the server. When the registration phase is completed, the server sends the public and private key-pair to the sender and receiver. Then, during data access, the receiver needs to send a request to the sender in the encrypted form. The sender sends the data in the encrypted form to the receiver after a successful authentication from the server. The sender does not ask the receiver for any personal information. So, it is not possible for the malicious user to get access to the required information from the receiver, and thus, the proposed scheme resists the phishing attack.

6. Complexity analysis

The time complexities of the encryption, decryption, key generation and key retrieval algorithms are computed for the proposed scheme and three other existing schemes, which are presented in Table 5.

Table 5 shows that the time complexities of the encryption and decryption algorithms of the proposed scheme is far better than the scheme proposed by Pramanik (Pramanik and Setua, 2004). Moreover, the proposed scheme has less time complexity for the key generation and key retrieval algorithms than the schemes proposed by Kaundal (Kaundal and Verma, 2015) and Paul (Paul et al., 2016). The overall time complexity of the proposed scheme is $O(n)$, which is better than the existing schemes.

7. Performance analysis

In this section, experimental setup and the results and discussions have been presented.

7.1. *Experimental setup*

The performance of the proposed system is evaluated in a Java framework installed on a Lenovo Core i5 Laptop with 8 GB RAM, 1 TB HDD, 256 GB SSD and Ubuntu 20.04 as the operating system. The Apache2 web server is used to implement the key-pair generator. The sender and receiver entities are implemented in Java and interfaced with the server using Java socket programming. The data used for evaluating the proposed scheme is taken from the 20 Newsgroup dataset (Lang, 2020), which is a popular dataset used in many research works. This dataset is selected as it contains textual data suitable for evaluating the proposed scheme.

7.2. *Results and discussions*

The proposed scheme is compared with the existing schemes proposed by Kaundal (Kaundal and Verma, 2015), Pramanik

Table 5 – Time complexity analysis.

Algorithm	Kaundal (Kaundal and Verma, 2015)	Pramanik (Pramanik and Setua, 2004)	Paul (Paul et al., 2016)	Proposed Scheme
Encryption	$O(n)$	$O(n^2)$	$O(n)$	$O(n)$
Decryption	$O(n)$	$O(n^2)$	$O(n)$	$O(n)$
Key generation	$O(n^2)$	$O(n)$	$O(n^2)$	$O(n)$
Key retrieval	$O(n^2)$	$O(n)$	$O(n^2)$	$O(n)$

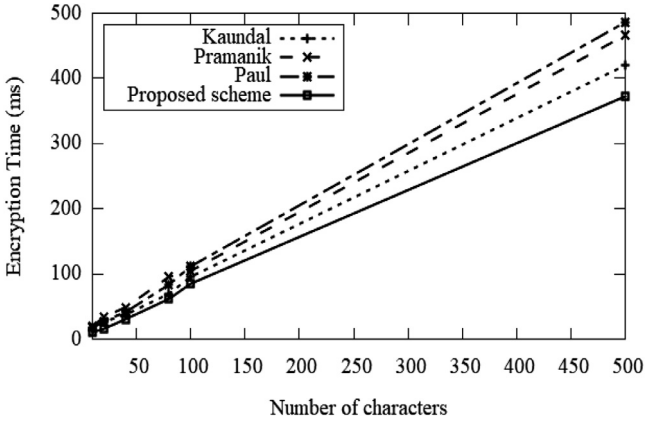

Fig. 3 – Encryption time vs. plaintext.

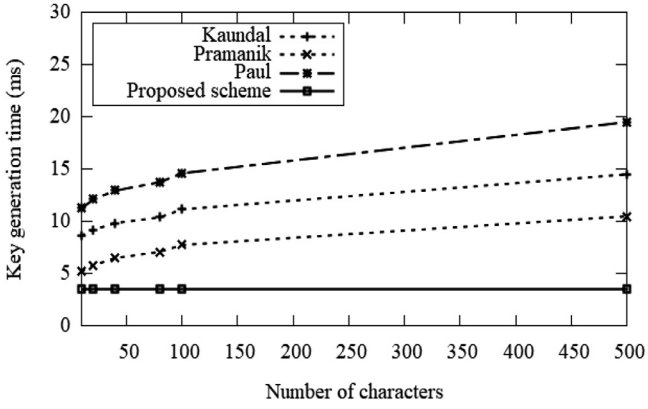

Fig. 5 – Key generation time vs plaintext.

(Pramanik and Setua, 2004) and Paul (Paul et al., 2016). The scheme proposed by Paul (Paul et al., 2016) has a limitation that it is not suitable for large messages and the key size is small. This scheme is mostly used for mobile devices, and it is not suitable for other configurations. Kaundal (Kaundal and Verma, 2015) has proposed a Feistel-inspired structure for DNA cryptography. An OTP is used in this scheme for key generation, and it increases encryption and decryption time. The DNA cryptography technique is based on DNA digital coding in the scheme proposed by Pramanik (Pramanik and Setua, 2004), and DNA hybridization is applied in this scheme. A critical issue of this scheme is that it is computationally complex.

The encryption and decryption times are evaluated based on six sets of data taken from the dataset, which are classified based on their size. The time taken for encryption and decryption are computed by performing 50 experiments for each classification of inputs. Finally, the average values have been considered from the results, which can be shown in Table 6. The encryption and decryption times of the existing and proposed schemes are compared, and it is noticed that the proposed system is better in terms of encryption and decryption times even using different data sizes. The results are presented in the graphs given in Figs. 3 and 4. It is observed that the encryption time in the proposed scheme is reduced by 24% and the decryption time is reduced by 80%.

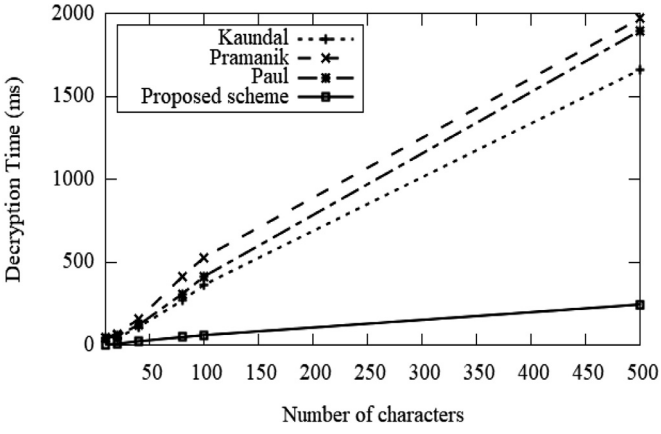

Fig. 4 – Decryption time vs. plaintext.

Table 6 – Time taken for encryption and decryption.

Plaintext length (char)	Kaundal (Kaundal and Verma, 2015)		Pramanik (Pramanik and Setua, 2004)		Paul (Paul et al., 2016)		Proposed Scheme	
	Encryption Time (ms)	Decryption Time (ms)	Encryption Time (ms)	Decryption Time (ms)	Encryption Time (ms)	Decryption Time (ms)	Encryption Time (ms)	Decryption Time (ms)
10	15	37	20	49	18	42	11.31	5.87
20	27	44	35	68	28	57	16.42	9.57
40	38	110	49	159	42	123	31.11	24.75
80	70	270	96	414	84	310	62.27	49.11
100	96	365	105	530	112	415	85.45	61.25
500	420	1660	466	1768	486	1896	312.45	245.75

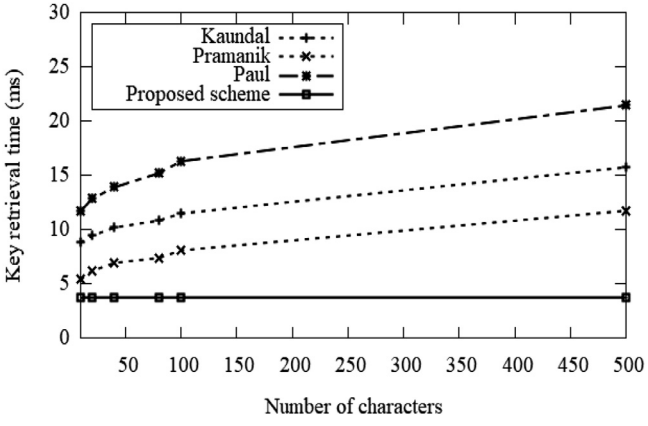

Fig. 6 – Key retrieval time vs. plaintext.

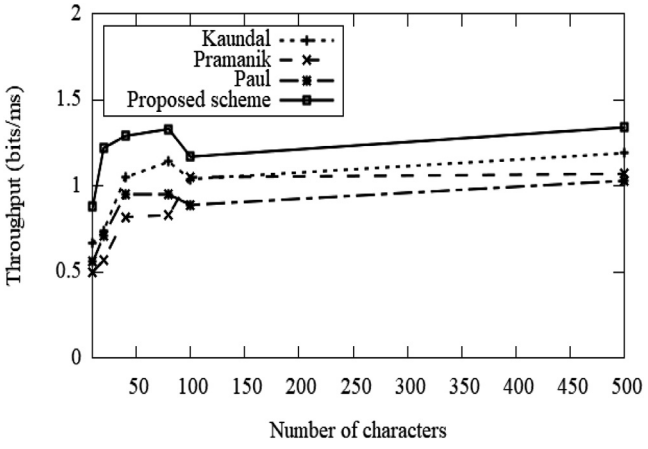

Fig. 7 – Throughput of the different schemes.

The key generation time and key retrieval time are calculated for the proposed scheme and three other existing schemes (Kaundal and Verma, 2015; Pramanik and Setua, 2004; Paul et al., 2016) for performance evaluation. The results are given in Table 7. In the proposed scheme, the sender collects receiver's attributes, which are subjected to simple operations, such as concatenation, ASCII conversion and binary conversion to form a 256-bit key. In the scheme proposed by Pramanik (Pramanik and Setua, 2004), the key is generated as an OTP that is represented as a DNA string. The length of the key depends on the length of the plaintext message. Here, each bit of the plaintext is replaced with 10 bases of a DNA sequence. Hence, the size of the key is 10 times the size of the binary plaintext message. In (Kaundal and Verma, 2015), a novel approach has been proposed based on DNA cryptography using feistel inspired structure. In this approach, a random key sequence based on OTP is generated that uses pseudo-random generator and a seed of 32 bytes DNA sequence as input. Similar to the scheme proposed by Pramanik (Pramanik and Setua, 2004), each bit of the plaintext is replaced with 5 bases of a DNA sequence. Hence, the size of the key is 5 times the size of the binary plaintext message. Therefore, the schemes proposed by Pramanik (Pramanik and Setua, 2004) and Kaundal (Kaundal and Verma, 2015) take more time for key generation. In (Paul et al., 2016), the sender and receiver agree on two large prime numbers. Many multiplications and modulus operations are performed in this scheme to generate the symmetric key, which is the number of DNA bases presented in a DNA sequence. The length of the final key is multiplied by the length of the plaintext. Therefore, computations in all these schemes are more complex than the proposed scheme. Fig. 5 shows the results of the key generation time.

Fig. 6 shows the results of the key retrieval time. In the proposed scheme, the receiver uses his/her attributes and executes simple operations, such as concatenation, ASCII conversion and binary conversion on the attributes to retrieve the 256-bit key. In the algorithm proposed by Paul (Paul et al., 2016), more complex operations are involved and there are a number of multiplications and modulus operations to retrieve the key. In both the schemes proposed by Kaundal (Kaundal and Verma, 2015) and Pramanik (Pramanik and Setua, 2004), the size of the ciphertext is almost 10 times of the binary plaintext. The key retrieval time is more in these existing scheme because of executing many complex operations. Thus, in all the existing schemes, the key retrieval is more than the proposed scheme.

The throughput of the proposed scheme and the existing schemes are evaluated based on the data size and encryption time. The results are given in Table 8. It is easily understand-

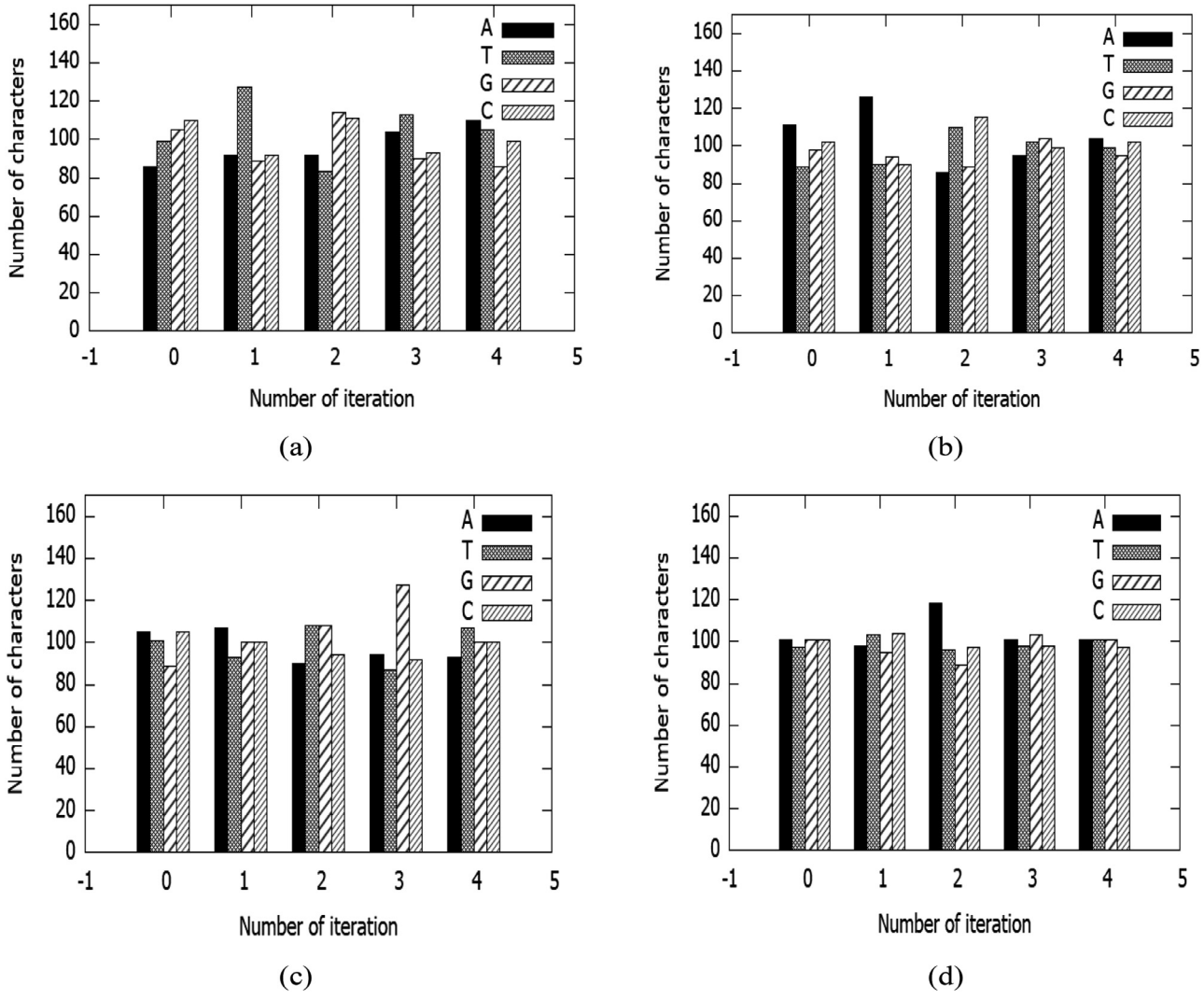

Fig. 8 – Frequency of DNA bases in ciphertext. (a) Plaintext with 100 A's (b) Plaintext with 100 T's (c) Plaintext with 100 G's (d) Plaintext with 100 C's.

Table 7 – Time taken for key generation and key retrieval.

Plaintext length (char)	Kaundal (Kaundal and Verma, 2015)		Pramanik (Pramanik and Setua, 2004)		Paul (Paul et al., 2016)		Proposed Scheme	
	Key generation Time (ms)	Key retrieval Time (ms)	Key generation Time (ms)	Key retrieval Time (ms)	Key generation Time (ms)	Key retrieval Time (ms)	Key generation Time (ms)	Key retrieval Time (ms)
10	8.65	8.85	5.25	5.45	11.25	11.75	3.52	3.75
20	9.20	9.45	5.80	6.20	12.10	12.85	3.52	3.75
40	9.85	10.20	6.50	6.95	12.95	13.91	3.52	3.75
80	10.40	10.85	7.10	7.40	13.75	15.20	3.52	3.75
100	11.15	11.50	7.75	8.10	14.60	16.31	3.52	3.75
500	14.50	15.75	10.50	11.75	19.50	21.45	3.52	3.75

Table 8 – Throughput (bits/ms) of the different schemes.

Plaintext length (char)	Kaundal (Kaundal and Verma, 2015) Throughput (bits/ms)	Pramanik (Pramanik and Setua, 2004) Throughput (bits/ms)	Paul (Paul et al., 2016) Throughput (bits/ms)	Proposed Scheme Throughput (bits/ms)
10	0.67	0.5	0.56	0.88
20	0.74	0.57	0.71	1.22
40	1.05	0.82	0.95	1.29
80	1.14	0.83	0.95	1.33
100	1.04	1.05	0.89	1.17
500	1.19	1.07	1.03	1.6

Table 9 – Frequency of DNA bases in the ciphertext for 100 A's.

Input: Plaintext of 100 A's				
Iteration	Frequency of occurrence of DNA bases in ciphertext			
	A	T	C	G
1	86	99	105	110
2	110	105	86	99
3	92	83	114	111
4	104	113	90	93
5	92	127	89	92

Table 10 – Frequency of DNA bases in the ciphertext for 100 T's.

Input: Plaintext of 100 T's				
Iteration	Frequency of occurrence of DNA bases in ciphertext			
	A	T	C	G
1	111	89	98	102
2	126	90	94	90
3	86	110	89	115
4	95	102	104	99
5	104	99	95	102

Table 11 – Frequency of DNA bases in the ciphertext for 100 G's.

Input: Plaintext of 100 G's				
Iteration	Frequency of occurrence of DNA bases in ciphertext			
	A	T	C	G
1	105	101	89	105
2	107	93	100	100
3	90	108	108	94
4	93	107	100	100
5	94	87	127	92

able from Fig. 7 that the throughput achieved in the proposed scheme is better than the existing schemes.

An analysis is done by using a plaintext of 100 characters made of only one DNA base. The frequency of occurrence of each DNA base in the ciphertext is found, and a few sample values are obtained, which are represented in Tables 9-12. The histogram distribution displayed in Fig. 8(a)–(d) clearly indicates that all the DNA bases have a significant presence in the ciphertext, which show the efficiency of the proposed scheme.

Avalanche effect is a cryptographic property, which states that a small change in the plaintext or the secret key results a significant change in the ciphertext. To analyse this effect in the proposed scheme, 1-bit of the 256-bit key is flipped and the results are tabulated in Table 13.

Table 12 – Frequency of DNA bases in the ciphertext for 100 C's.

Input: Plaintext of 100 C's				
Iteration	Frequency of occurrence of DNA bases in ciphertext			
	A	T	C	G
1	101	97	101	101
2	98	103	95	104
3	118	96	89	97
4	101	98	103	98
5	101	101	101	97

Table 13 – Changes in ciphertext for 1-bit change in key.

Plaintext	Change in the ciphertext
100 A's	72%
100 T's	85%
100 G's	82%
100 C's	79%

The results show that when 1-bit of the secret key is flipped, the minimum change in the DNA bases of the ciphertext is 72% and the maximum change is 85%. Thus, it has been proved that the proposed scheme is secured and efficient.

8. Conclusions and future works

A novel cryptosystem is proposed in this paper for secure data transmission from the sender to the receiver. In the proposed scheme, a 256-bit secret key is generated, and the plaintext undergoes various cryptographic operations and converted into a DNA sequence. A randomly generated Mealy machine is used for coding DNA sequence, thereby making the ciphertext more secure. The proposed scheme supports security against various security attacks. Experimental results show that the proposed system performs with better encryption time, decryption time, key generation time and key retrieval time when compared with the existing systems. The proposed scheme is likewise better than the existing schemes as far as the throughput of the system is considered. The avalanche effect is also successful as a small change in the secret key can make a significant change in the ciphertext. Hence, the proposed DNA based cryptosystem is efficient and secure than the existing systems. The security of the proposed scheme may be demonstrated mathematically in the future.

Credit statement

Pramod Pavithran is the main author of this paper, who has conceived the idea and discussed it with all co-authors.

Sheena Mathew has performed the complexity analysis and simulations of this paper.

Suyel Namasudra is the corresponding author, who has performed the security analysis and write up of this work.

Pascal Lorenz has supervised the entire work, evaluated the performance and security analysis and proof-read the paper.

Declaration of Competing Interest

None.

REFERENCES

Alguliyev RM, Aliguliyev RM, Sukhostat LV. Efficient algorithm for big data clustering on single machine. CAAI Trans. Intell. Technol. 2020;5(1):9–14.

Baykara M, Gürel ZZ. Detection of phishing attacks. In: 6th International Symposium on Digital Forensic and Security (ISDFS); 2018. p. 1–5. doi:10.1109/ISDFS.2018.8355389.

Chang X, Yan A, Zhang H. Ciphertext-only attack on optical scanning cryptography. Opt. Lasers Eng. 2020;126. doi:10.1016/j.optlaseng.2019.105901.

Clelland CT, Risca V, Bancroft C. Hiding messages in DNA microdots. Nature 1999;399(6736):533–4.

Devi D, Namasudra S, Kadry S. A boosting-aided adaptive cluster-based undersampling approach for treatment of class imbalance problem. Int. J. Data Warehous. Min. (IJDWM) 2020;16(3):60–86.

Gehani A, LaBean T, Reif J. DNA-based cryptography. In: Jonoska N, Paun G, Ozenberg G, editors. In: Aspects of Molecular Computing. Springer; 2000. p. 167–88.

Gupta R, Singh RK. An improved substitution method for data encryption using DNA sequence and CDMB. In: Proceedings of the 3rd International Symposium; 2015. p. 197–206. doi:10.1007/978-3-319-22915-7_19.

Hopcroft JE, Motwani R, Ullmann JD. Introduction to Automata theory, languages, and Computation. 3rd Ed. Pearson Education India; 2008.

Hossain EMS, Rokibul AKM, Biswas MR, Morimoto Y. A DNA cryptographic technique based on dynamic DNA sequence table. In: Proceedings of the IEEE International Conference on Computer and Information Technology (ICCIT). Bangladesh: IEEE; 2016. p. 270–5.

Hsu HZ, Lee RCT. DNA based encryption methods. In: Proceedings of the 23rd Work-Shop on Combinatorial Mathematics and Computation Theory; 2006. p. 145–50.

Kalyani S, Gulati N. Pseudo DNA cryptography technique using OTP key for secure data transfer. Int. J. Eng. Sci. Comput. 2016;6(5):5657–63.

Kaundal AK, Verma AK. Extending feistel structure to DNA cryptography. J. Discr. Math. Sci. Cryptogr. 2015;18(4):349–62. doi:10.1080/09720529.2014.995975.

Kumar S, Wollinger T. Fundamentals of symmetric cryptography. In: Lemke K, Paar C, Wolf M, editors. In: Embedded Security in Cars. Springer; 2006. p. 125–43.
Lang, K, 2020. 20 Newsgroups. Available: http://qwone.com/~jason/20Newsgroups/ [Accessed on 13 August 2020].
Leier A, Richter C, Banzhaf W, Rauhe H. Cryptography with DNA binary strands. Biosystems 2000;57(1):13–22.
Li S, Wang G, Yang J. Survey on cloud model based similarity measure of uncertain concepts. CAAI Trans. Intell. Technol. 2019;4(4):223–30.
Namasudra S. Taxonomy of DNA-based security models. In: Namasudra S, Deka GC, editors. In: Advances of DNA Computing in Cryptography. Taylor & Francis; 2018b. p. 53–68.
Namasudra S. Cloud computing: a new era. J. Fundam. Appl. Sci. 2018a;10(2):113–35.
Namasudra S. An improved attribute-based encryption technique towards the data security in cloud computing. Concurr. Comput. 2019;31(3). doi:10.1002/cpe.4364.
Namasudra S. Data access control in the cloud computing environment for bioinformatics. Int. J. Appl. Res. Bioinform. (IJARB) 2020b;11(1). doi:10.4018/IJARB.2021010105.
Namasudra S. Fast and secure data accessing by using DNA computing for the cloud environment. IEEE Trans. Serv. Comput. 2020a. doi:10.1109/TSC.2020.3046471.
Namasudra S, Chakraborty R, Kadry S, Manogaran G, Rawal BS. FAST: fast accessing scheme for data transmission in cloud computing". Peer-to-Peer Netw. Applica. 2020b. doi:10.1007/s12083-020-00959-6.
Namasudra S, Chakraborty R, Majumder A, Moparthi NR. Securing multimedia by using DNA based encryption in the cloud computing environment. ACM Trans. Multim. Comput. Commun. Applica. 2020a;16(3s). doi:10.1145/3392665.
Namasudra S, Deka GC. Applications of Blockchain in Healthcare. Springer; 2021. doi:10.1007/978-981-15-9547-9.
Namasudra S, Deka GC, Bali R. Applications and future trends of DNA computing. In: Namasudra S, Deka GC, editors. In: Advances of DNA Computing in Cryptography. Taylor & Francis; 2018. p. 181–92.
Namasudra S, Deka GC, Johri P, Hosseinpour M, Gandomi AH. The revolution of blockchain: state-of-the-art and research challenges". Arch. Comput. Meth. Eng. 2020c. doi:10.1007/s11831-020-09426-0.
Namasudra S, Devi D, Kadry S, Sundarasekar R, Shanthini A. Towards DNA based data security in the cloud computing environment. Comput. Commun. 2020d;151:539–47.
Namasudra S, Roy P. Time saving protocol for data accessing in cloud computing. IET Commun. 2017;11(10):558–1565.
Namasudra S, Roy P, Vijayakumar P, Audithan S, Balamurugan B. Time efficient secure DNA based access control model for cloud computing environment. Future Gener. Comput. Syst. 2017;73:90–105.
Namasudra S, Sharma S, Deka GC, Lorenz P. DNA computing and table based data accessing in the cloud environment. J. Netw. Comput. Appl. 2020e;172. doi:10.1016/j.jnca.2020.102835.
Nath Nayak G, Samaddar SG. Different flavours of Man-In-The-Middle attack, consequences and feasible solutions. In: 3rd International Conference on Computer Science and Information Technology; 2010. p. 491–5. doi:10.1109/ICCSIT.2010.5563900.
Paul S, Anwar T, Kumar A. An innovative DNA cryptography technique for secure data transmission. Int. J. Bioinform. Res. Appl. 2016;12(3). doi:10.1504/IJBRA.2016.078235.
Pramanik S, Setua SK. DNA cryptography. In: Proceedings of the 7th IEEE International Conference on Electrical & Computer Engineering. Bangladesh: IEEE; 2004. p. 551–4.
Rama R, Suyambu JB, Andrew A, Shanmugam S. A study of DES algorithm with cellular automata. Int. J. Innov. Manag. Inform. Product. 2012;3(1):10–16.
Salamatian S, Huleihel W, Beirami A, Cohen A, Médard M. Why botnets work: distributed brute-force attacks need no synchronization. IEEE Trans. Inf. Forensics Secur. 2019;14(9):2288–99.
Sarkar M, Saha K, Namasudra S, Roy P. An efficient and time saving web service based android application. SSRG Int. J. Comput. Sci. Eng. 2015;2(8):18–21.
Schneier B. Applied Cryptography. 2nd ed. Wiley & Sons; 1996.
Sharma P, Lal N, Diwakar M. Text security using 2D cellular automata rules. In: Proceedings of the Conference on Advances in Communication and Control Systems. Atlantis Press; 2013. p. 263–368.
Soni R, Prajapati G, Khan A, Kulhare D. Triple stage DNA cryptography using sequential machine. Int. J. Advan. Res. Comput. Sci. Softw. Eng. 2013;3(8).
Sundaram GS, Pavithra S, Arthi A, Bala BM, Mahalakshmi S. Cellular automata based DNA cryptography algorithm. *Proceedings of the 9th International Cconference on Intelligent Systems and Control* (ISCO). India: IEEE, 2015.
Tornea O, Borda ME. Security and complexity of a DNA-based cipher. In: Proceedings of the 11th Roedunet International Conference. Romania: IEEE; 2013. p. 1–5.
Wang X, Zhang Q. DNA computing-based cryptography. In: Proceedings of the 4th IEEE International Conference on Bio-Inspired Computing; 2009. p. 1–3.
Wang Z, Yu Z. Index- based symmetric DNA encryption algorithm. In: Proceedings of the 4th International Congress on Image and Signal Processing. Shanghai: IEEE; 2011. p. 2290–4.
Yuan Y, Mo MY. Security for cyber-physical systems: secure control against known-plaintext attack. Sci. China Technol. Sci. 2020;63:1637–46.
Zhang S, Gao T. A novel data hiding scheme based on DNA coding and module-N operation. Int. J. Multim. Ubiqui. Eng. 2015;10(4):337–44.
Zhang Y, Fu B, Zhang X. DNA cryptography based on DNA fragment assembly. In: Proceedings IEEE International Conference on Information Science and Digital Content Technology. South Korea: IEEE; 2012. p. 179–82.
Zhao H, Han G, Wang L, Wang W. MILP-based differential cryptanalysis on round-reduced midori64. IEEE Access 2020;8:95888–96.
Zhao X, Li R, Zuo X. Advances on QoS-aware web service selection and composition with nature-inspired computing. CAAI Trans. Intell. Technol. 2019;4(3):159–74.

1. Mr. Pramod Pavithran
Division of Computer Engineering, School of Engineering
Cochin University of Science and Technology
Kochi, Kerala
India.

Biography: **Pramod Pavithran** received his B. Tech. degree in computer science and engineering from Kerala University, Thiruvananthapuram, India and M. Tech. in computer and information science from Cochin University of Science and Technology, Kochi, India. Currently, he is an associate professor in School of Engineering, Cochin University of Science and Technology. His-research interests include cryptography, network security and theory of computation.

2. Prof. Sheena Mathew
Division of Computer Engineering, School of Engineering
Cochin University of Science and Technology
Kochi, Kerala
India.

Biography: **Prof. Sheena Mathew** received her PhD degree in computer science from Cochin University of Science and Technology, Kochi, India. She is currently a professor in School of Engineering, Cochin University of Science and Technology. She has over

27 years teaching experience and has more than 50 research papers in referred journals and proceedings. Her research interests include network security, natural language processing and text-mining.

3. Dr. Suyel Namasudra (Corresponding Author)
Department of Computer Science and Engineering
National Institute of Technology Patna
Patna, Bihar
India.

Biography: **Dr. Suyel Namasudra** is an Assistant Professor in the Department of Computer Science and Engineering at the National Institute of Technology Patna, Bihar, India. Prior to joining the National Institute of Technology Patna, Dr. Namasudra was an Assistant Professor in the Department of Computer Science Engineering at the Bennett University, India. He has received PhD in Computer Science and Engineering from National Institute of Technology Silchar, Assam, India. His-research interests include Cloud Computing, DNA Computing and Information Security. Dr. Namasudra has edited 2 books, 1 patent and 35 publications in refereed journals, book chapters and conference proceedings. He has participated in many international conferences as an Organizer and Session Chair. Dr. Namasudra is an Editorial Board member and a Reviewer of many journals.

4. Prof. Pascal Lorenz
University of Haute-Alsace
Colmar, Alsace-Champagne-Ardenne
France.

Biography: **Pascal Lorenz** received his M.Sc. (1990) and Ph.D. (1994) from the University of Nancy, France. Between 1990 and 1995 he was a research engineer at WorldFIP Europe and at Alcatel-Alsthom. He is a professor at the University of Haute-Alsace, France, since 1995. His-research interests include QoS, wireless networks, high-speed networks and DNA Computing. He is the author/co-author of 3 books, 3 patents and 200 international publications in refereed journals and conferences.

He was Technical Editor of the IEEE Communications Magazine Editorial Board (2000–2006), IEEE Networks Magazine since 2015, IEEE Transactions on Vehicular Technology since 2017, Chair of IEEE ComSoc France (2014–2018), Financial chair of IEEE France (2017–2019), Chair of Vertical Issues in Communication Systems Technical Committee Cluster (2008–2009), Chair of the Communications Systems Integration and modeling Technical Committee (2003–2009), Chair of the Communications Software Technical Committee (2008–2010) and Chair of the Technical Committee on Information Infrastructure and Networking (2016–2017). He has served as Co-Program Chair of IEEE WCNC'2012 and ICC'2004, Executive Vice-Chair of ICC'2017, TPC Vice Chair of Globecom'2018, Panel sessions co-chair for Globecom'16, tutorial chair of VTC'2013 Spring and WCNC'2010, track chair of PIMRC'2012 and WCNC'2014, symposium Co-Chair at Globecom 2007–2011, Globecom'2019, ICC 2008–2010, ICC'2014 and '2016. He has served as Co-Guest Editor for special issues of IEEE Communications Magazine, Networks Magazine, Wireless Communications Magazine, Telecommunications Systems and LNCS. He is Associate Editor for International Journal of Communication Systems (IJCS-Wiley), Journal on Security and Communication Networks (SCN-Wiley) and International Journal of Business Data Communications and Networking, Journal of Network and Computer Applications (JNCA-Elsevier).

He is a senior member of IEEE, IARIA Fellow and member of many international program committees. He has organized many conferences, chaired several technical sessions and gave tutorials at major international conferences. He was IEEE ComSoc Distinguished Lecturer Tour during 2013–2014.